DTU Library

Spacio-temporal situation assessment for mobile robots

Beck, Anders Billesø; Risager, Claus; Andersen, Nils Axel; Ravn, Ole

Published in:
Proceedings of the fourteenth International fusion conference on Information Fusion

Publication date:
2011

Document Version
Publisher's PDF, also known as Version of record

Spacio-Temporal Situation Assessment for Mobile Robots

Anders Billesø Beck
Claus Risager
Centre for Robot Technology
Danish Technological Institute
Odense, Denmark
Email: [anbb, cr]@dti.dk

Nils A. Andersen
Ole Ravn
DTU Electrical Engineering
Technical University of Denmark
Kgs. Lyngby, Denmark
Email: [naa, or]@elektro.dtu.dk

Abstract—**In this paper, we present a framework for situation modeling and assessment for mobile robot applications. We consider situations as data patterns that characterize unique circumstances for the robot, and represented not only by the data but also its temporal and spacial sequence. Dynamic Markov chains are used to model the situation states and sequence, where stream clustering is used for state matching and dealing with noise. In experiments using simulated and real data, we show that we are able to learn a situation sequence for a mobile robot passing through a narrow passage. After learning the situation models we are able to robustly recognize and predict the situation.**

Keywords: Automated Situation Awareness, Markov Models, Clustering, Streaming data.

I. Introduction

Mobile robots face the challenge of operating in environments that are dynamic and often undefined for the robot. A fundamental requirement for robust long term operation, is that the robot are able comprehend its internal states and states of the environment into a situation assessment. This situation awareness can be used to take informed decisions, prevent errors or simply choose control strategies for better and robust performance. Representing and assessing situations are not a trivial task, as situations are not only characterized by a snapshot of information, but also by the temporal relations and sequences.

Examples of situations for mobile robots can be the sequence of docking to a loading area, passing a human in an otherwise empty hallway or passing through a narrow passage, such as a door. Most of these situations can be anticipated by internal states of the controller but the situation assessment can act as a bird's perspective i.e. to predict if the robot is moving towards states that are known to be problematic.

The goal of this work is to present a modeling framework, that can be used to detect situations and errors, which might only be roughly known before the robot is set into operation or even unexpected situations that were difficult anticipate in the application planning. In this paper, we show how our framework can be applied to recognize an example situation of passing through a door, using both an imposed and a learned model, to demonstrate the flexibility.

We present a framework for on-line situation assessment on mobile robot systems using the first order Extensible Markov Model (EMM), proposed by Dunham et al. [1] as structural back-end. EMM can efficiently represent spacio-temporal situation models, without having complex relationships between observations and states as the hidden Markov model [2]. Unique for the presented approach is that the modeling framework continuously learns new states, even in the application phase. This allows the operator to classify situation signatures as they emerge or review crash sequences to set-up future crash prediction models.

This paper contributes with a practical approach to modeling, tracking and predicting situations for a robotic system. Although the framework is focused towards mobile robotics, it has a general applicability to a wide range of domains, such as industrial automation systems for predicting possible production stops and detecting rare events such as sensor failures. Also within Human Robot Interaction situation assessment would also be a very useful tool to assess interaction situations and possibly predict intentions.

The paper is organized as follows. The next section reviews related work within the area. Section III describes our approach for situation modeling and on-line data processing. Finally Section IV present experimental evaluation of our framework, using both simulated and real data.

II. Related Work

Situation awareness (SA) has been a major topic for decades among human factors engineers when evaluating human decision making in dynamic environments. Dr. Mica Endsley have pioneered the analytical area of this field and presented countless publications on practical evaluation of operator Situation Awareness. Endsley also defines a model for situation awareness, that have three levels of awareness [3]:

1) Perception of elements in the environment

978-0-9824438-2-8/11/$26.00 ©2011 IEEE

2) Comprehension of their meaning
3) Projection of their future status

Any process to achieve, acquire and maintain situation awareness in this framework is referred to as situation assessment. Endsley's model has formed the background for our situation model, where we fuse environment information and model their relations. Future status predictions are possible through state transition probabilities of our model.

Information pattern recognition for multidimensional data are often approached using fundamental algorithms such as k-means or incorporating probabilistics using Gaussian mixtures and the expectation maximization (EM) algorithm [4]. Both algorithms are very useful for extracting information clusters from finite data sets, but unsuited for the sequential nature of data streams. Most significantly do these approaches all assume that data points are Independent and Identically Distributed (IID), where sensor inputs gathered from a robot is often a time dependent continuous stream. When data are encoded in a stream, there is often important information encoded in the spacio-temporal patterns of the data, such as correlations between data that are close in sequence.

One of the strongest tools for analyzing spacio-temporal information is the Markov chain. The traditional approach to Makov chains assumes a finite structure and number of states. Dunham et al. [1] confronts this limitation with the Extensible Markov Model (EMM) that are described in more detail in section III-A. Dynamic extensions to the Markov Model, somewhat similar to the EMM have been presented by Goldberg and Mararić [5] as the Augmented Markov Models (AMM). Like EMM, AMM can be considered as a degenerate Hidden Markov Model [2], with an observation probability of 1.0 for a particular symbol and 0.0 the others. In practice, this removes the hidden state assumption. Where the AMM can grow during a learning cycle and then remains fixed at runtime, EMM continues to grow even at runtime. Furthermore, where AMM treats each measurement of different value as a unique state, EMM applies clustering to handle data noise and reduce the state-space.

Clustering information streams have received a fair bit of attention recently for processing the constantly growing volume of available real-time data such as web click-streams. Traditional approaches for clustering streams have been the nearest neighbor algorithm, but noteworthy algorithms such as BIRCH [6] and STREAM [7] have specifically designed for the computational requirements of clustering continuous data streams, without the need for scanning the entire datasets and even being capable of integrating measurement noise (BIRCH).

Recent work by Meyer-Delius et al. [8], [9], [10] present a framework using Hidden Markov Models as a base for modeling and online-recognizing situations in traffic scenarios. This approach has many benefits from the formal definitions of HMM and the associated well-founded probabilistic models for dealing with sensor noise. However it is limited to pre-imposed situation models where data matching must be pre-learned into the HMM and becomes static in the recognition phase. It makes the approach less suitable for detecting new situations using incomplete models or adapting to changing situations at runtime.

III. Situation Model

Modeling complex real-world scenarios such as situations are a complex task. A situation is not only characterized by a set of circumstances at a given time, but also by the sequence of these circumstances. So we have naturally looked into the graph-based domain for suitable models. Secondly, we have emphasized our efforts in a modeling scheme, where the structure and relations are interpretable, graphically representable and even understandable for the advanced user.

In opposition to other related work, such as Meyer-Delius et al. [8], we have chosen a slightly simpler approach for modeling situations. Using a first order Markov chain, we can represent the spatio-temporal nature of situations together with transition probabilities, without the complexity of hidden states in a HMM. States in the Markov chain are a unique set of information or circumstances at one given time. We define a situation instance as the sequence of one or more of these states, who can be translated to meaningful knowledge about the state of a dynamic environment.

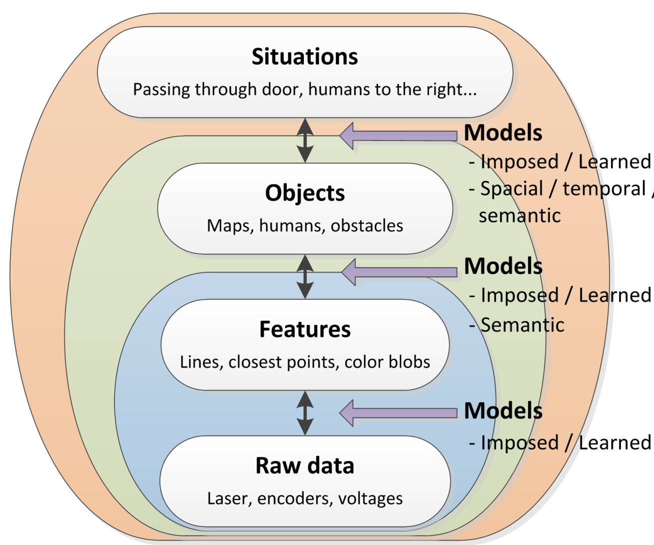

Figure 1. Perception information for mobile robots are often organized in a hierarchy. Situations combine these layers in the spacio-temporal dimension.

Situations add an additional level of abstraction to traditional models, as symbolic and sub symbolic information can be fused to form the situation states and combine them in the temporal and spacial dimension. Figure 1 illustrate a mobile robot perception hierarchy where situation models are placed in the highest level, as an assessment can involve data from all preceding layers.

Following Endsley's SA model [3], a comprehensive situation model should be capable of covering the three levels illustrated in Figure 2. Our model has three similar layers of abstraction, which are a natural part of the Markov chain approach.

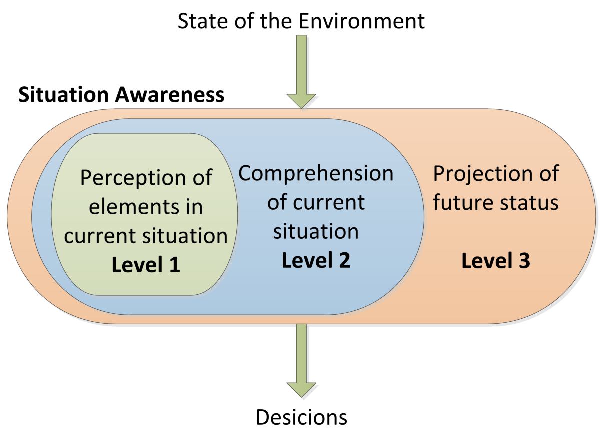

Figure 2. Endsleys models of Situation Awareness defines three levels of awareness.

Level 1 models
defines what data that should be included in the model. Data selection must be done carefully, as too many parameters can cloud the observed structures and too little might give inconclusive observations.

Level 2 models
defines specific state-sequences that represent a situation. These models can be pre-imposed or classified from learned data during runtime.

Level 3 models
evaluates the transition likelihoods from the current state to other defined states. These models are used for prediction of future error states and inference i.e. for triggering counter measures.

In the presented work, we focus primarily on designing a level 1 model and show through experiments that it can be used to learn meaningful level 2 situation models and level 3 prediction models.

Examples of situations that can be modelled using data from the various perception layers are:

- Detection and validation of motion patterns using raw encoder feedback and motor speed commands
- Detection of passage through a door using distance and angle features extracted from laserscanner

A. Extensible Markov Model

Markov Chains (MC) and its variations such as Finite State Automatons are one of the most powerful tools for analyzing and modeling complex system structures, behaviors or data relations. MC has the benefit of being rather simple and easily representable graphically, but has key problems when modeling real dynamic environments:

- The model might be incomplete or not existing at construction time
- Environments changes over time so should the model.

Dunham et al. deal with these problems in the EMM framework [1], a first order MC model that continuously is able to learn new states during the application (prediction, etc.) phase. Furthermore, the EMM are able to merge or delete nodes that no longer belong to the model and transition probabilities for each edge are maintained when new states are added or removed.

The state space of the model is represented by

$$S = \{s_1, ...s_m\} \tag{1}$$

with a set of m states at a given time where

$$A = \begin{vmatrix} a_{00} & \cdots & a_{i0} \\ \vdots & \ddots & \vdots \\ a_{0j} & \cdots & a_{ij} \end{vmatrix} \tag{2}$$

is the state transition matrix where a_{ij} represents the transition likelihood from stage s_i to s_j and

$$N = \begin{vmatrix} n_{00} & \cdots & n_{i0} \\ \vdots & \ddots & \vdots \\ n_{0j} & \cdots & n_{ij} \end{vmatrix} \tag{3}$$

represent the number of transitions from state s_i to s_j . The transition likelihood from state s_i to s_j are maintained online by computing the number of outbound transitions from s_i to s_j in relation to the total number of outbound transitions from s_i, given by

$$L(s_t = s_j | s_{t-1} = s_i) = \frac{n_{ij}}{\sum_{z=1}^{m} n_{iz}} \tag{4}$$

Figure 3 illustrate how EMM handles state transitions, when a new state can either be matched to an existing state or a new state are created.

Valid transition likelihoods are important for predicting occurrence of future, important states and hereby approach Level 3 Situation Awareness. The EMM has further advantages for applications in situation modeling and assessment. State matching are based on a data clustering, where a model independent clustering algorithm determines the state relationship of measurements, instead of learning data association into the models as with Hidden Markov Models.

B. Data Clustering

Forming a useful MC, which represents essential states in the state-space, relies on the clustering algorithm. When a new sensor data-set are received, the clustering algorithm decides if the data-set fit within an existing cluster or a new

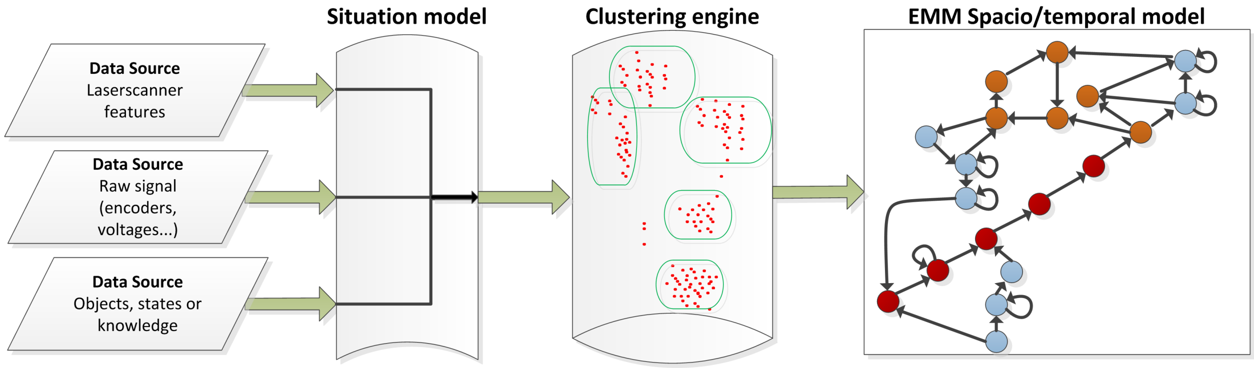

Figure 4. The basic elements of the situation assessment architecture. Information from many data sources can be fused by the situation model. The clustering engine groups and matches the data in clusters, which are structured in the spacio-temporal domain by the EMM model.

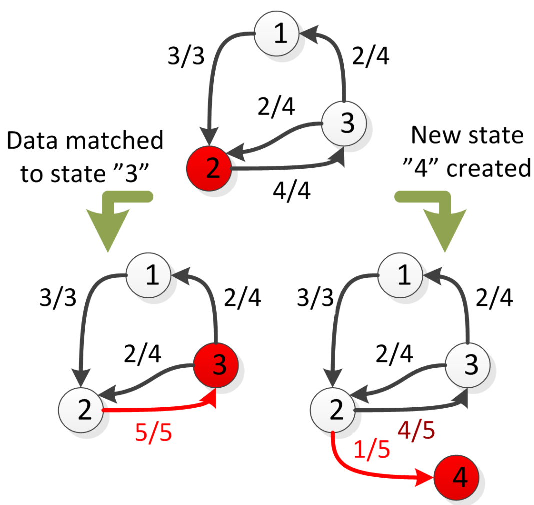

Figure 3. Example of state transition for EMM when data can be matched to an existing cluster and when a new cluster must be created. Transition probabilities are maintained accordingly

cluster and state must be created.

Our initial implementation uses the nearest neighbor clustering algorithm, and the Jaccard similarity coefficient used as similarity metric

$$J(s_i, s_j) = \frac{|s_i \cap s_j|}{|s_i \cup s_j|} \tag{5}$$

A static similarity threshold, λ, are used to determine if a data-set matches en existing cluster or a new cluster should be created.

The clustering algorithm is the key filter for dealing with sensor noise as well as maintaining the compromise between too few and too many states. Future work has already started for evaluating more complex streaming clustering algorithms such as BIRCH and STREAM.

C. Architecture

Mobile robotic systems are often heterogeneous systems, with sensor data from many sources and even more internal structures of perception algorithms, estimation filters and controllers. To fit a situation assessment system to this environment, n architecture have been designed for the situation assessment system i Java, containing the following elements:

- **Data source model** using a flexible plug-in based structure and supporting any number of real-time data providers.
- **Situation model** using a XML-based loader that defines the situation state-space and using a event driven subscription to data providers.
- **Clustering engine**, which handles the state matching and data reduction.
- **EMM spacio-teporal model** for recognition and prediction of states

Figure 4 illustrate the architecture components for assessment of one situation model. Situation (level 1) models are defines the set of data variables that are processed. Possible noisy sensor data or discrete information are then clustered to reduce the state-space and to generalize the observations. Finally the clusters are organized in the EMM structure, where sequences can be classified and transition likelihoods estimated.

Markov chains are very illustrative for humans when they are visualized as directed graphs. A directed graph shows the spatio-temporal properties of the model as well as general structure. We use the 3D graph visualization tool Gephi [11] that are designed directly for visualizing and exploring large graphs. Furthermore Gephi also supports online streaming of graphs and can be used as user interface to recognize the current states, render the state graph while it is evolving and examine the state relations. Figure 5 show an on-line generated state graph, that were produced during the simulation tests in section IV-A. The relevant states have been manually classified, identifying the green states leading into the detected situation, the red states representing the actual situation and the blue states leaving the situation.

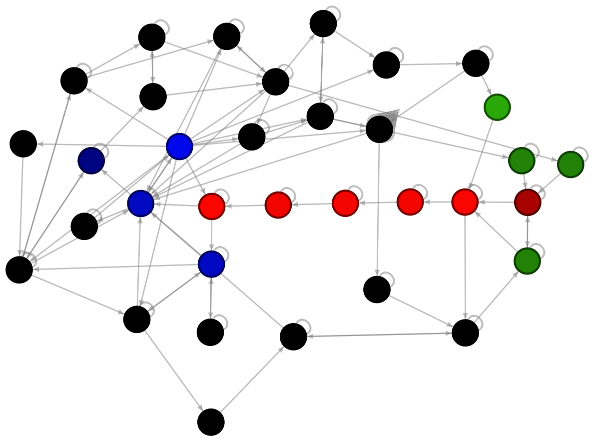

Figure 5. Gephi visualization of a Markov chain generated during the simulated evaluation tests. States are manually classified and color-coded for on-line identification

IV. Experimental evaluation

Evaluation of our framework have been done using the DTU Mobotware mobile robot control framework [12]. We have considered a classical problematic scenario for mobile robots: Passage through narrow passages, such as doors or between obstacles. Especially for medium sized robots (50-80 cm wide) passage through doors often quite difficult and require constrained control strategies.

A level 1 situation model has been designed to uniquely identify the characteristics of passing through a narrow space. A laserscanner are used to measure the distances to the nearest obstacle at each side of the robot, as illustrated on figure 6

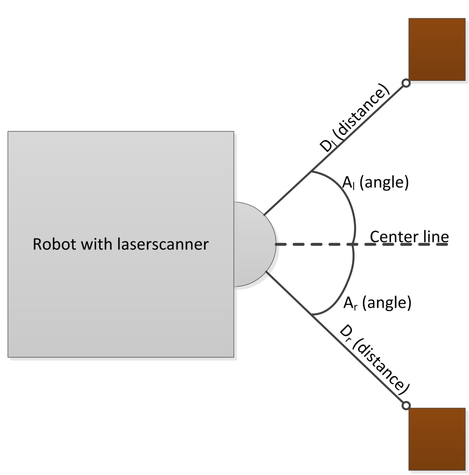

Figure 6. Laserscanner is used to detect the closest feature to each side of the robot.

Two features are extracted from each measurement, where D_l and D_r are distances to the closest point on left and right side of the robot centre line. A_l and A_r are the angles between the D_l and D_r vectors and the centre line. For detecting the unique situation signature of moving towards and through a narrow passage, the model uses the sum of the two closest obstacles and the difference of the angles.

$$s_n = \begin{cases} d = D_l + D_r & \text{Sum of distances} \\ a = A_l - A_r & \text{Difference of angles} \end{cases} \tag{6}$$

This model is invariant for the relative position inside the passage, as the distances simply are summed. Using the difference of angles as the other feature, makes the robot invariant for the relative orientation in the passage, as the angle difference will remain the same. Tests have shown this feature set to be sufficient to give a robust recognition of narrow space passage.

A. Simulated narrow passage detection

Using the Stage 3.2.2 simulator [13] we performed a quantitative test to investigate the characteristics of the narrow space detection model. The goal of the experiment was to investigate how tolerant the detector was against variations in approach angle and placement within the narrow passage. Furthermore, we wanted to investigate the possibilities of tracking and prediction of the situation.

A simulation of 20 passages through a 60 cm wide door by a 30 cm wide robot with random starting points and orientations was carried out. The simulation was initialized without any predefined states, so the first pass through the passage learned the states, which were used for matching the remaining simulated runs. A Jaccard similarity threshold of $\lambda = 0.99$ was used for cluster matching in the experiment. For comparing the situation assessment results from the 20 simulated passes, we plot the physical trajectories of the robot during the simulation and contours of the states that were learned and detected at each location in figure 8. For plotting purposes, state information has been interpolated between the robot trajectories, causing some noisy edge effects around the top and bottom of the plot.

Each colour layer represents a region where the robot is estimated to be the same state. Clearing the plot from robot trajectories in figure 9 clarifies the state transition diagrams. Observe how the situation assessment follows a similar sequence of states for each pass of the robot, except for a few miss-matches around the (-0.6,0.2) region. When the robot have passed through the door, the state sequence gets less predictable, as the laserscanner now see the closest obstacles on the other side of the passage.

Already from a significant distance, the state sequence predicts a high the likelihood of a narrow passage. For example, observe in figure 9 how the state sequence is almost linear from 60 cm before the door and until it has been passed.

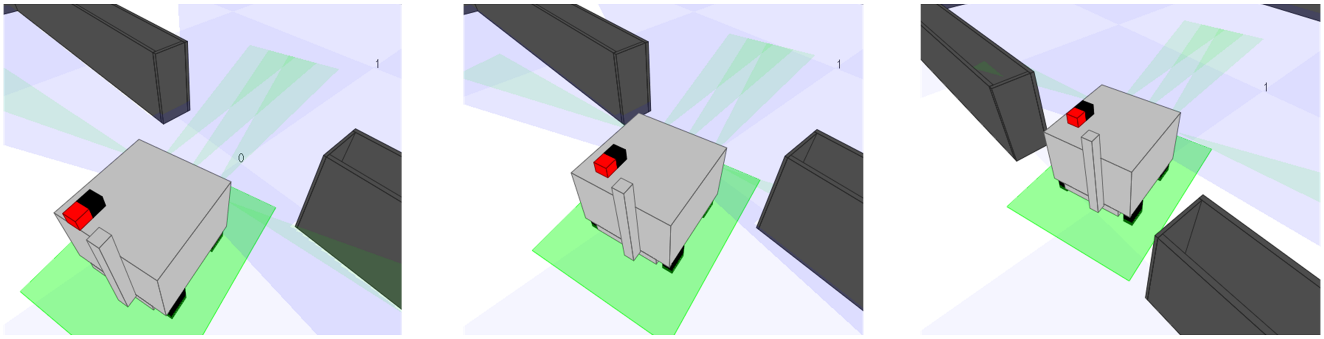

Figure 7. Simulated passage through a doorway repetitively and approaching from various angles

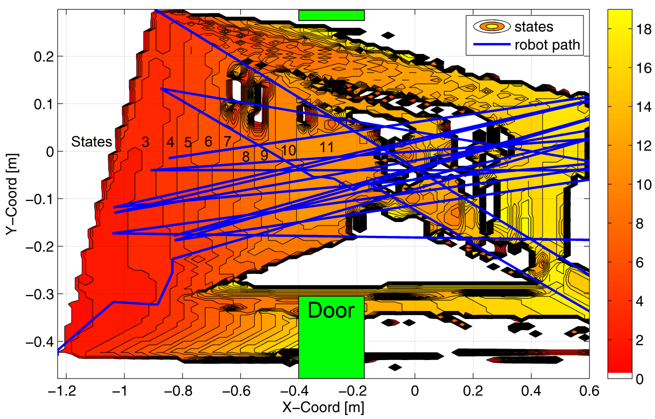

Figure 8. Visualization of the robot trajectories and situation state regions. Region noise are primarily caused by plotting interpolation between measurement points rather than state-matching noise. White areas are regions that has not been classified within the 18 first samples and have been removed for plotting purposes.

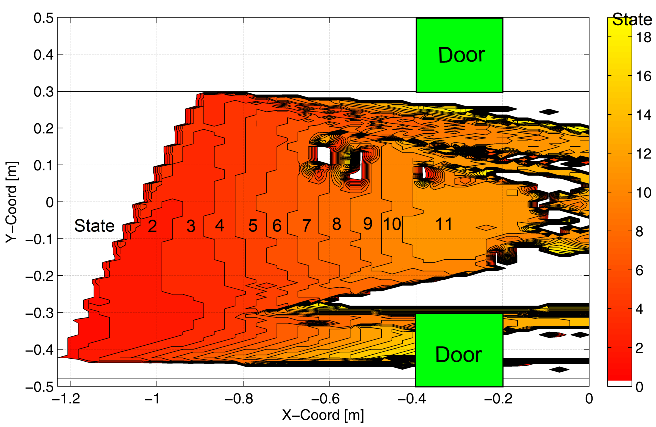

Figure 9. Close-up on state regions without the robot trajectories. It is clearly evident that each robot pass have followed a similar sequence, except for a few spurious states around (-0.6,0.2) that has caused steep gradients in the interpolation.

State 11 is observed to be the centre of the door, with a state data-set of

$$s_{11} = \begin{cases} d = 0.61 \text{ m} \\ a = 3.09 \text{ rad} \end{cases} \tag{7}$$

where the combined distance from the laserscanner equals the width of the door, and the angle difference between the closest points are 180°. Following the state transitions up to state 11 in the state transition likelihood matrix reveals a clearly identifiable sequence back to state 4. To make a robust model, we have chosen a level 2 situation model to represent the narrow passage by the following state sequence X_s:

$$X_s = \{7, 8, 9, 10, 11\} \tag{8}$$

The transition likelihoods for the states of X_s are shown in table I.

Table I
STATE TRANSITION LIKELIHOODS FOR THE STATES OF THE LEVEL 2 SITUATION MODEL X_s THROUGHOUT THE SIMULATION.

State	Transition to	Likelihood
7	8	1.00
8	9	0.95
	7	0.05
9	10	1.00
10	11	0.91
	12	0.09
11	12	0.55
	14	0.05
	34	0.30

The state transition table I can also be visualised in a directed graph, where the likelihood of predicting a passage through the narrow space is clearer.

The typical sequence have been identified by inspection as of figure 10, so the prediction accuracy can be calculated from the Markov chain by its maximum likelihood, given by

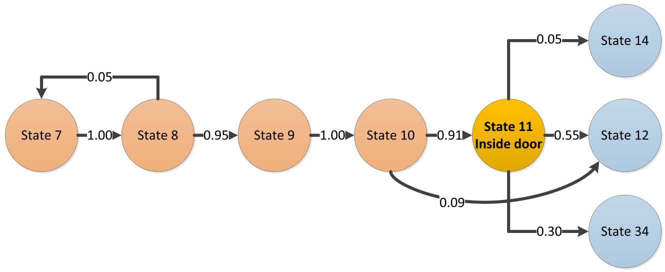

Figure 10. Summary of the state transition likelihood for passing through the door. Through the 20 simulated runs, prediction through the chain is very high.

$$L(s_{11}|s_7) = \prod_{i \in X_s} a_{i,i-1} = 0.86 \tag{9}$$

which can be considered as a firm ground for changing to a motion planner, optimized for careful navigation through the passage. In opposition to similar work for example by Meyer-Delius et al. [8], both situations and state space are learned from actual operation in the environment. By the flexibility of our approach, situations and states can be classified during on-line operation, pre-imposed by hand-crafted models or trained from a data-set.

In this experiment, we have shown that a level 1 model can be defined without too much complexity to produce meaningful situation states. By classification, we have extracted a level 2 model X_s, that robustly detects the planned situation and finally shown that a level 3 model can be derived allowing us to use transition likelihoods for situation prediction.

B. Evaluation using real world data

Using a small mobile robot platform, the framework was also evaluated in a real world setup. The robot is similar to the simulated robot in figure 7 and is also running the DTU MobotWare mobile robot control software. For sensing the narrow passage through the door, the robot was equipped with a Hokuyo URG-04LX laserscanner mounted 5 cm above the ground, configured to cover 180 ° in front of the robot. The scanner samples the distance for every 0.35 ° at a frequency of 10 Hz. Using this set-up, the robot was navigated through two laboratory doors and one office door.

Like in the simulated tests, the first passage through a laboratory door generated the states that subsequently were used for matching. A Jaccard similarity threshold of $\lambda = 0.99$ was used for cluster matching to keep results comparable to the simulated experiment. Unfortunately it is not possible to use the geographic position of the robot as a comparison measure in this test. Figure 11 shows the state transition timeline for passing through the doors.

Samples 10 to 18 on figure 11 (in red) are the actual passage through the doorway, that are clearly identified in

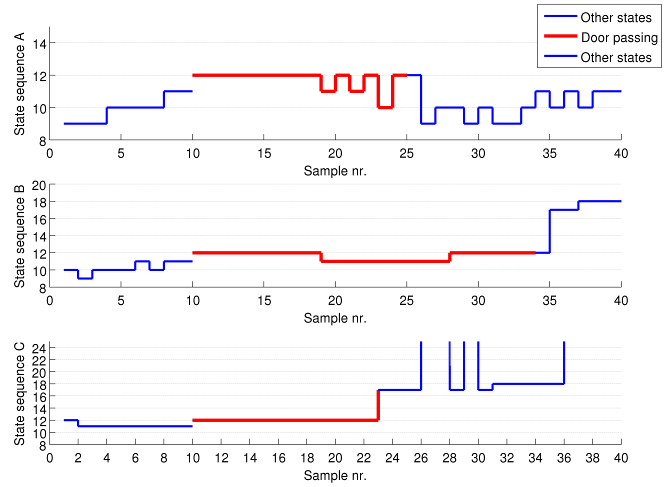

Figure 11. State transition timeline for passing through two lab doors (A and B) and an office door (C) in a traditionally cluttered university environment.

all three situations as state 12. Some state noise is present, especially when the robot has passed through the doorway, as we also saw on the simulated tests.

Very noteworthy is that state 12 are preceded by state 11 in all three cases. Examining the state transition matrix conclude that the likelihood for transition from state 11 to state 12 are

$$P(s_{12}|s_{11}) = 0.56 \tag{10}$$

Although the likelihood is not completely conclusive, it is still useful for predicting possible narrow passages ahead and i.e. for selecting a slow manoeuvre speed or a motion planner with focus on keeping the centre between obstacles. The last part of the red sections are passage past the open door and objects cluttering the doorway, creating noise but still some detections of state 11 and 12 indicating a narrow passage ahead or at the current locating.

In this experiment we have shown that a robust detection of being within a narrow passage is also possible using real data and parameters identical to the simulated experiment. Prediction are not possible through as many states as in the simulated experiment, but performance might be improved by optimizing the similarity threshold or selecting a more advanced clustering strategy that handles sensor and environment noise better.

V. Conclusion

In this work, we presented a framework for situation assessment for mobile robotic systems. To embrace the complexity of fusing information in these heterogeneous mobile robot control systems, we have designed an architecture that can combine data from the various sources and assess the situations it forms. Inspired from human situation awareness we have proposed a three layered model to characterize the situations and have applied the EMM as tool for on-line modeling and learning.

The approach has been evaluated experimentally using both simulated and real data, evaluating the performance of assessing the situation of a small mobile robot passing through a narrow passage. Our results demonstrate that the system are capable of learning states that identifies such a situation and subsequently use the learned model to robustly identify the situation. Additionally, we showed that our model can be used to on-line predict the likelihood for future occurrence of the learned situations.

VI. Future work

Many paths are still open for future work on this topic. Clustering of the streaming data has a significant impact on the learned states and could benefit from being implemented using more sophisticated and hierarchical algorithms such as BIRCH.

Also further work on fusion of pre-imposed states and states learned through clustering and automatic recognition of dominant sequences will further contribute to making situation assessment an important tool for solving detection, prediction and comprehension problems for autonomous systems.

References

[1] M. Dunham, Y. Meng, and J. Huang, "Extensible markov model," in *Fourth IEEE International Conference on Data Mining*, 2004, pp. 371–374.

[2] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, Issue 2, pp. 257–286, 1989.

[3] M. Endsley, "Towards a theory of situation awareness in dynamic systems," *Human Factors*, vol. 37, pp. 32–64, 1995.

[4] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society*, vol. Series B, pp. 1–38, 1977.

[5] D. Goldberg and M. J. Mataric, "Augmented markov models," Computer Science Department, University of Southern California, Tech. Rep., 1999.

[6] T. Zhang, R. Ramakrishnan, and M. Livny, "Birch: an efficient data clustering method for very large databases," in *International Conference on Management of Data*. ACM New York, NY, USA, 1996, pp. 103 – 114.

[7] S. Guha, N. Mishra, R. Motwani, and L. O'Callaghan, "Clustering data streams," in *41st Annual Symposium on Foundations of Computer Science (FOCS 2000)*. IEEE, 2000, pp. 359–366.

[8] D. Meyer-Delius, C. Plagemann, and W. Burgard, "Probabilistic situation recognition for vehicular traffic scenarios," in *2009 IEEE International Conference on Robotics and Automation*, Kobe, Japan, May 12-17 2009, pp. 459–464.

[9] D. Meyer-Delius, J. Sturm, and W. Burgard, "Regression-based online situation recognition for vehicular traffic scenarios," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, St. Louis, USA, 2009.

[10] D. Meyer-Delius, C. Plagemann, and W. Burgard, "Probabilistic situation recognition and its application to vehicular traffic situations," in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, Kobe, Japan, 2009, to appear.

[11] M. Bastian, S. Heymann, and M. Jacomy, "Gephi: An open source software for exploring and manipulating networks," in *International AAAI Conference on Weblogs and Social Media*, 2009. [Online]. Available: http://www.aaai.org/ocs/index.php/ICWSM/09/paper/view/154

[12] A. B. Beck, N. A. Andersen, J. C. Andersen, and O. Ravn, "Mobotware - a plug-in based software framework for mobile robots," in *7th IFAC Symposium on Intelligent Autonomous Vehicles*, Lecce, Italy, 2010.

[13] R. T. Vaughan, "Massively multi-robot simulations in stage." *Swarm Intelligence*, vol. 2(2-4), pp. 189–208, 2008.